

# **Napredni razvoj programske podpore za web**

**- predavanja -  
2022./2023.**

---

## **7. Jednostranične web-aplikacije (1/3)**

# Creative Commons



- slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **prerađivati** djelo



- pod sljedećim uvjetima:

- **imenovanje:** morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno:** ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima:** ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



*U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela.*

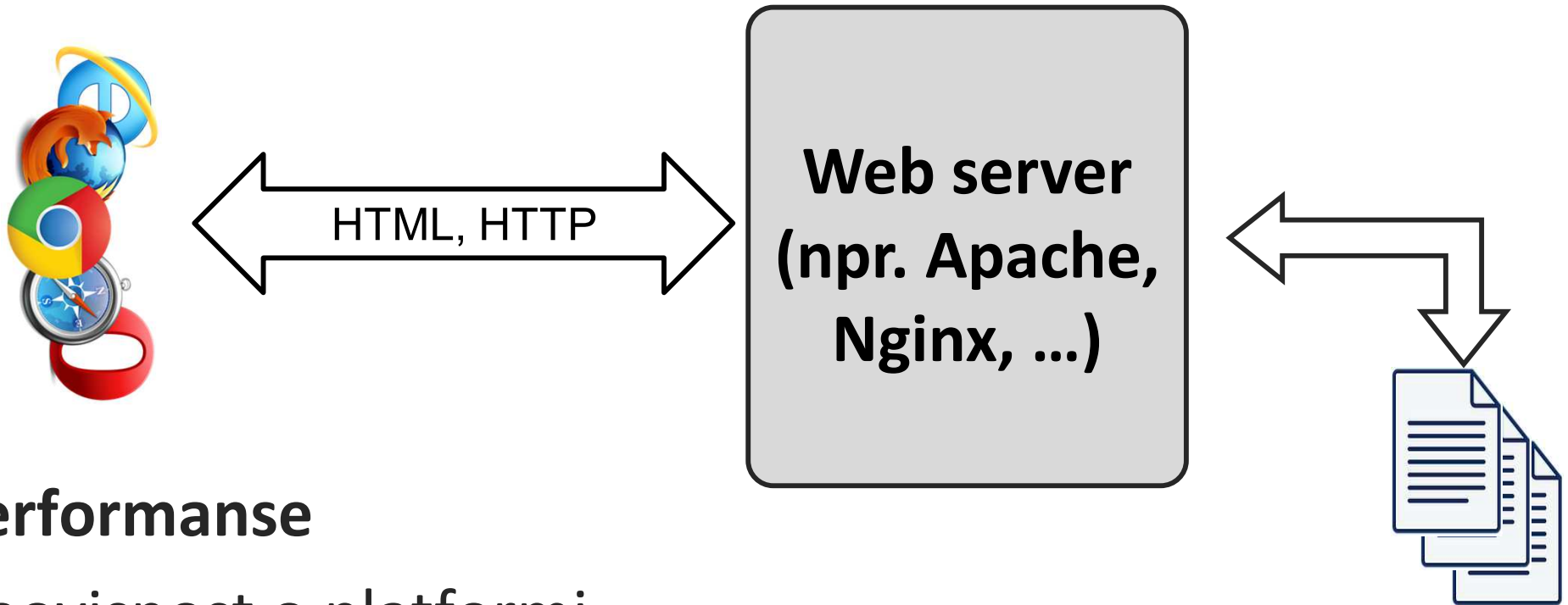
*Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.*

*Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.*

*Tekst licence preuzet je s <http://creativecommons.org/>*

# Podjetimo se – tri arhitekture web-aplikacija

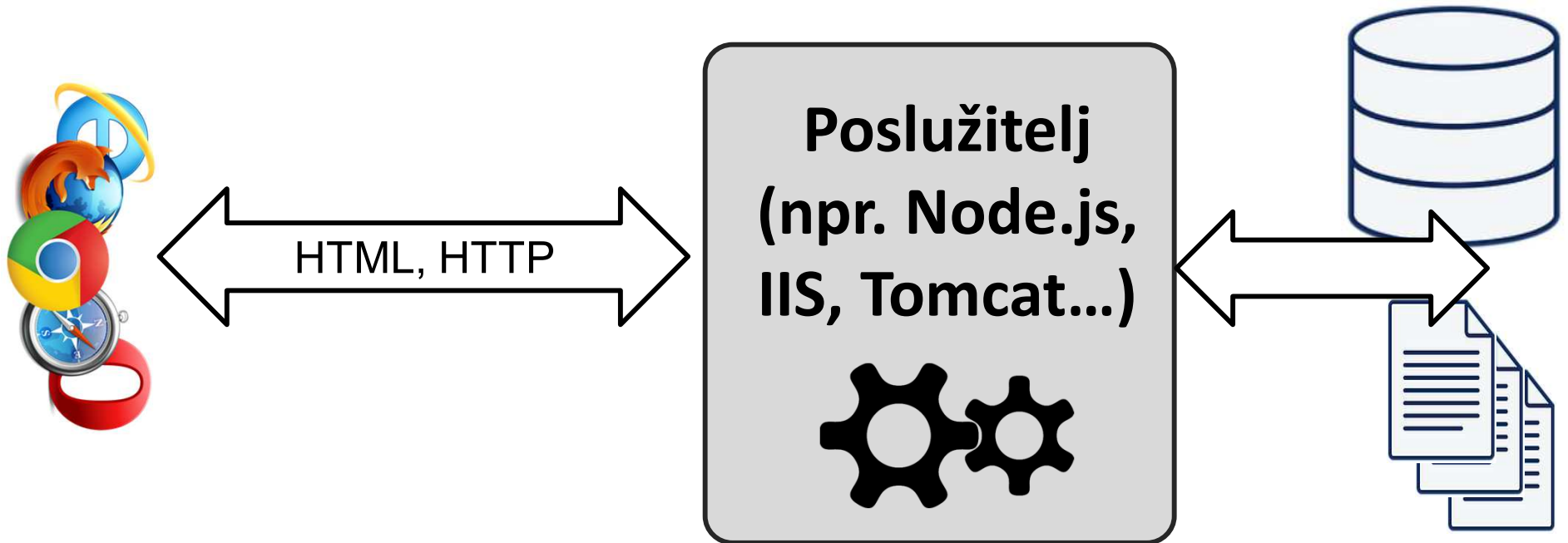
## 1. Statične web-stranice



- ✓ Performanse
- ✓ Neovisnost o platformi
- ✓ Sigurnost
- ✗ „Statičnost”, klijent mora osigurati bilo kakvu interaktivnost

# Podjetimo se – tri arhitekture web-aplikacija

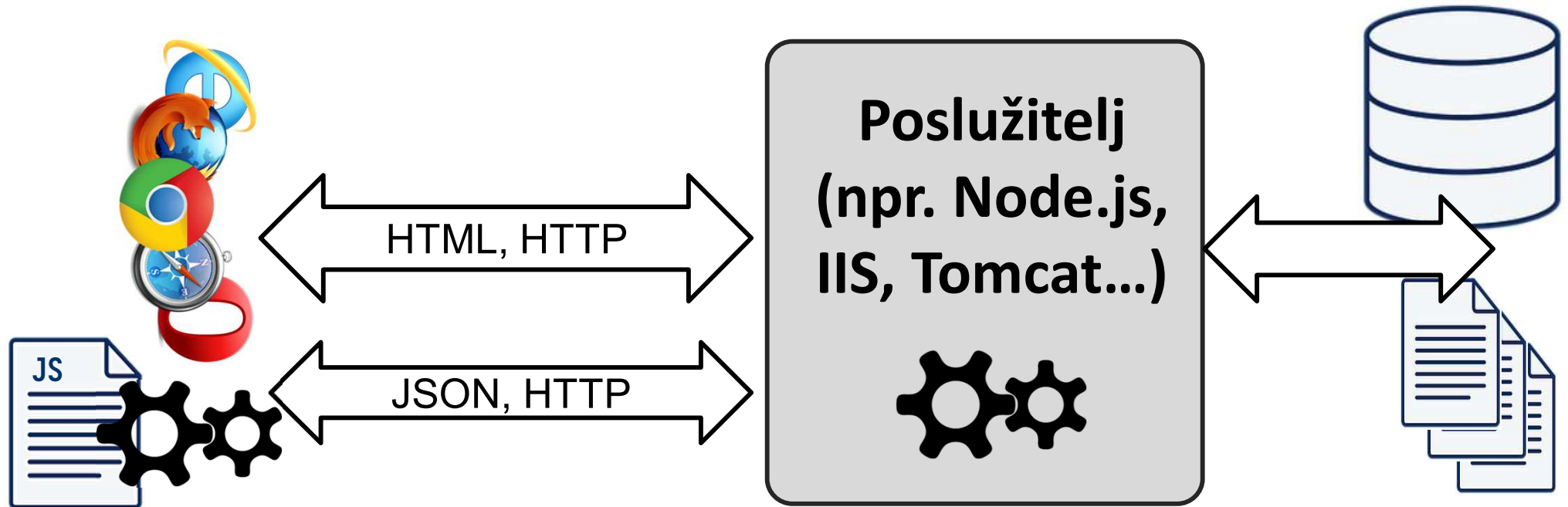
## 2. Dinamične web-stranice (generirane na poslužitelju)



- ✓ Dinamički sadržaj, prilagođen klijentu
- ✓ Sigurnost, odabir tehnologije, ...
- ✗ Procesiranje (opterećenje) na poslužitelju
- ✗ Responzivnost (UX): klijent predaje podatke i **čeka** odgovor

## 3. Jednostranične web-aplikacije

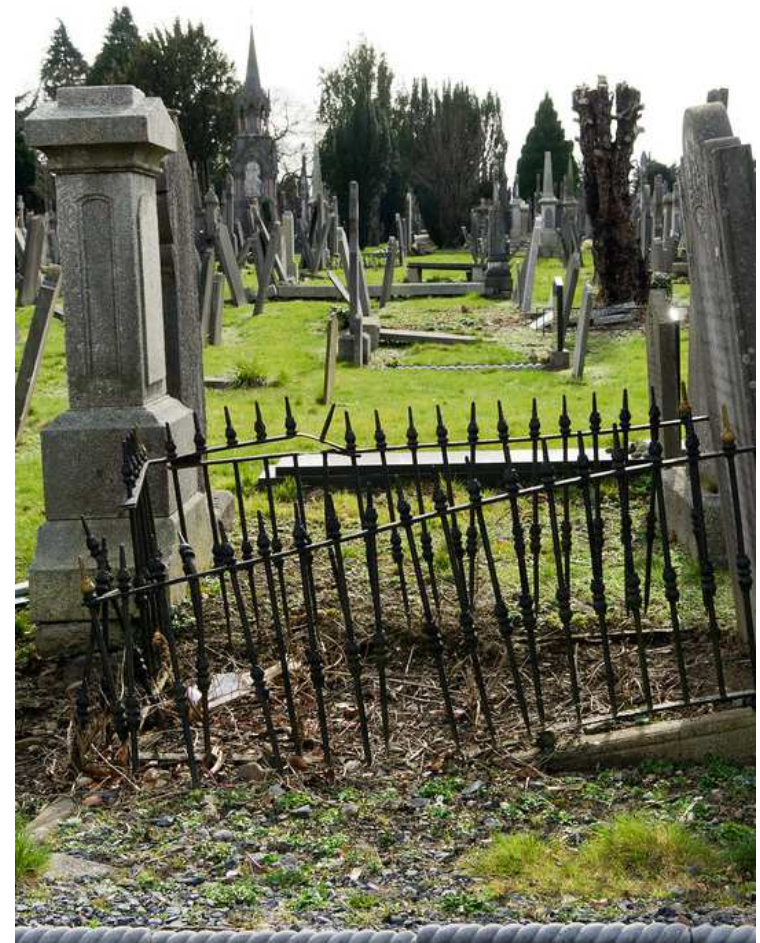
### ▪ SPA: *Single Page web Applications*



- ✓ **Dinamički sadržaj, prilagođen klijentu**
- ✓ **Performanse:** dobar dio opterećenja se prebacuje na klijenta
- ✓ **UX** – doima se kao (mobilna) aplikacija
- ✗ **Kompleksnost, teže testiranje, sigurnost, zadan JS**

# Kao i obično, ideja nije nova...

- **Java applets**
  - [https://en.wikipedia.org/wiki/Java\\_applet](https://en.wikipedia.org/wiki/Java_applet)
  - Java applets were introduced in the first version of the Java language, which was released in **1995**.
  - Beginning in 2013, [major web browsers began to phase out support for the underlying technology applets used to run](#), with applets becoming completely **unable to be run by 2015–2017**.
- **ActiveX**
  - <https://en.wikipedia.org/wiki/ActiveX>
  - Microsoft introduced ActiveX in **1996**
  - ActiveX is still supported as of Windows 10 through [Internet Explorer 11](#), while ActiveX is not supported in their default web browser [Microsoft Edge](#)
- **Flex/Flash**
  - [https://en.wikipedia.org/wiki/Apache\\_Flex](https://en.wikipedia.org/wiki/Apache_Flex)
  - In November **1996**, FutureSplash was acquired by Macromedia, and Macromedia re-branded and released FutureSplash Animator as Macromedia Flash 1.0.
  - Adobe donated Flex to the Apache Software Foundation in 2011[2] and it was promoted to a top-level project in December 2012.
- **Silverlight**
  - [https://en.wikipedia.org/wiki/Microsoft\\_Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight)
  - 2007-2011
- *It'd, ...razni JS kalkulatori, ...*
- **Novo: WASM** (<https://en.wikipedia.org/wiki/WebAssembly> )



As of March 2021, less than 0.01% of sites used Silverlight,<sup>[24]</sup> 2.1% used the discontinued Adobe Flash,<sup>[25]</sup> and less than 0.01% use Java (client-side; server-side 3.4% use Java).<sup>[26][27]</sup>  
[https://en.wikipedia.org/wiki/Microsoft\\_Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight)

# Primjer: $\mu$ SPA

---

**Izgradimo vlastiti mikro radni okvir za  
jednostranične web-aplikacije**

# ■ Zašto radimo svoj FWK koji ćemo baciti u smeće?

- Upoznati:
  - načela rada
  - mehanizme i tehnologiju
  - probleme, vidjeti što je teško
  - razumjeti
  - Možda i naučiti malo JS-a u procesu 😊
- Iz istog razloga što:
  - Učimo zbrajati i množiti na papiru
  - Učimo pokazivače u C-u da bi shvatili *by value, by reference*
  - Itd.

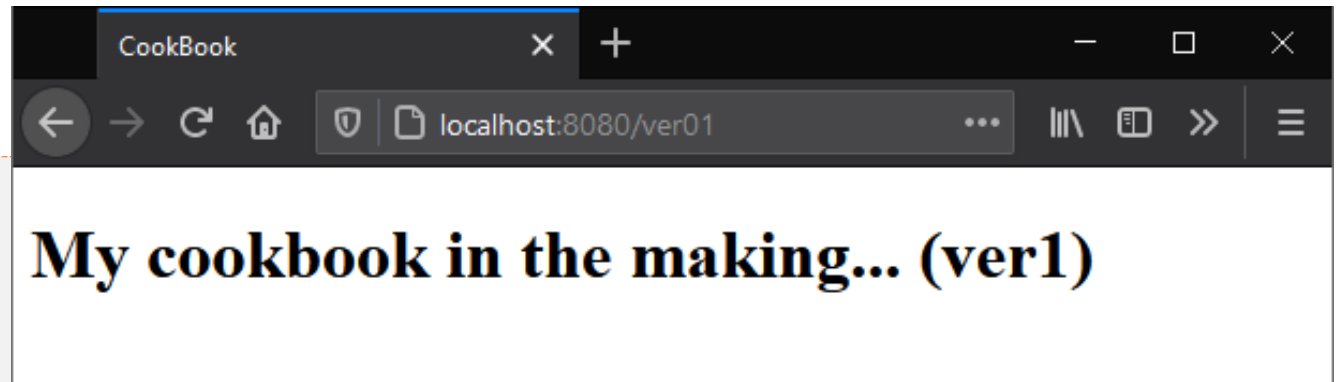


## src

- `git clone git@gitlab.com:fer-web2/lectures/src.git`

# Osnovno načelo: mijenjamo DOM iz JS koda

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CookBook</title>
</head>
<body>
  <div id="uspa-app"></div>
</body>
<script>
  document
    .getElementById("uspa-app")
    .innerHTML = "<h1>My cookbook in the making...</h1>";
</script>
</html>
```



# Digresija: server.js

- FWK razvijamo kroz 4 verzije – jedan poslužitelj koji poslužuje sve 4 verzije na izbor

Browser → :8080 → server.js

u tekućem direktoriju pronalazi sve direktorije koji počinju s "ver"

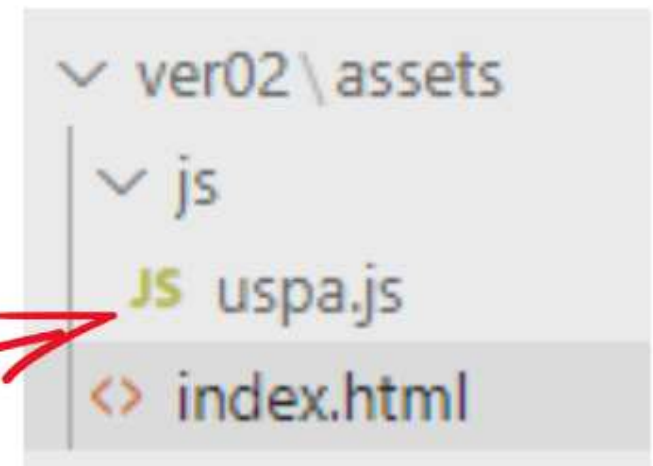
```
const getDirectories = (source) =>
  readdirSync(source, { withFileTypes: true })
    .filter(dirent => dirent.isDirectory() && dirent.name.startsWith("ver"))
    .map(dirent => dirent.name);
const arrVersionDirs = getDirectories(__dirname);
```

```
app.get('/', function (req, res) {
  res.send(`Please select version: <ul>${arrVersionDirs.map(name => `<li><a href="/${name}>${name}</a></li>`).join("")}</ul>`);
});
```

# Ver2: izmjestimo kod u modul

```
uspa > ver02 > assets > <> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  ... <meta charset="UTF-8">
5  ... <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  ... <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  ... <title>CookBook</title>
8  </head>
9  <body>
10 ... <div id="uspa-app">
11 ... </div>
12 </body>
13 <script type="module" src="/ver02/assets/js/uspa.js"></script>
14 </html>
```

novi ustroj projekta



ES modules

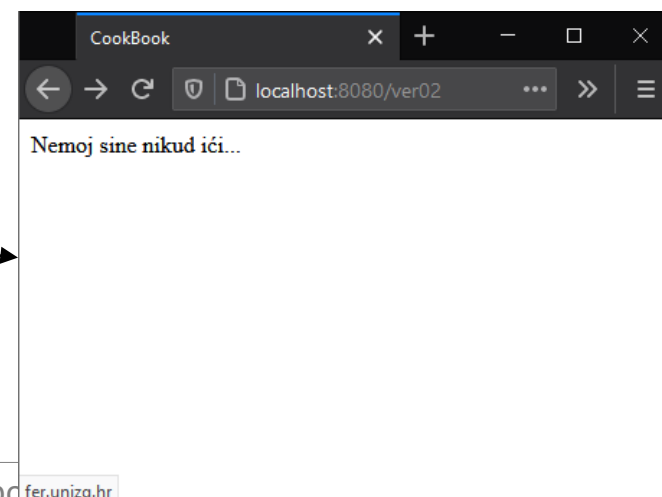
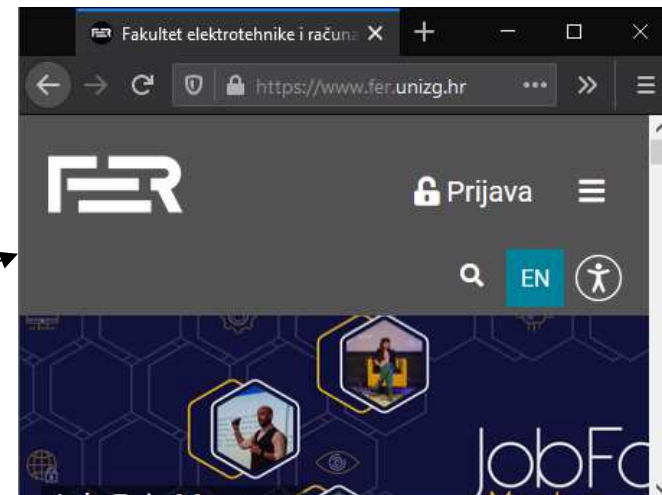
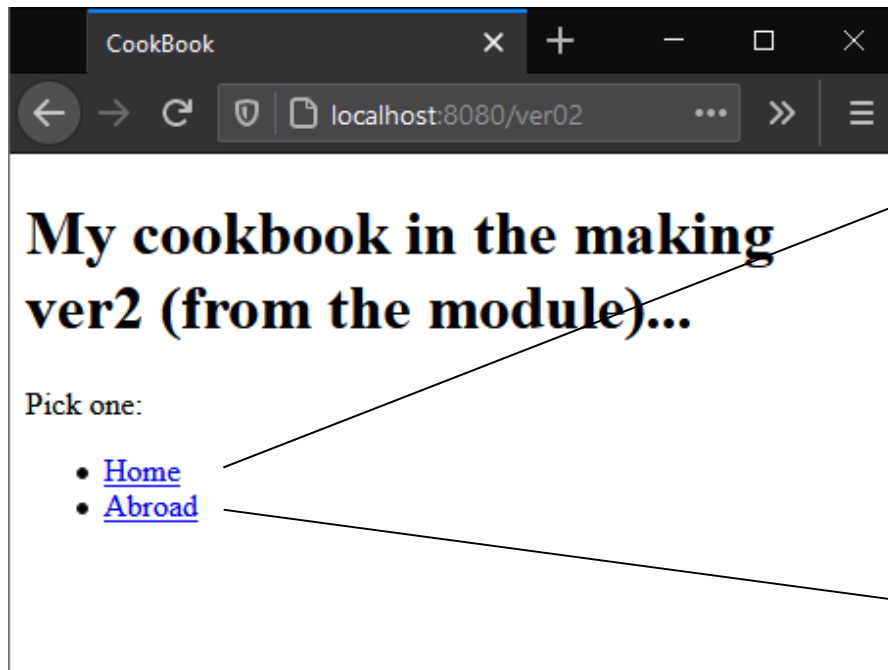
[https://en.wikipedia.org/wiki/ECMAScript#6th Edition %E2%80%93 ECMAScript 2015](https://en.wikipedia.org/wiki/ECMAScript#6th_Edition_%E2%80%93_ECMAScript_2015)

# Ver2: presrećemo svaki klik!

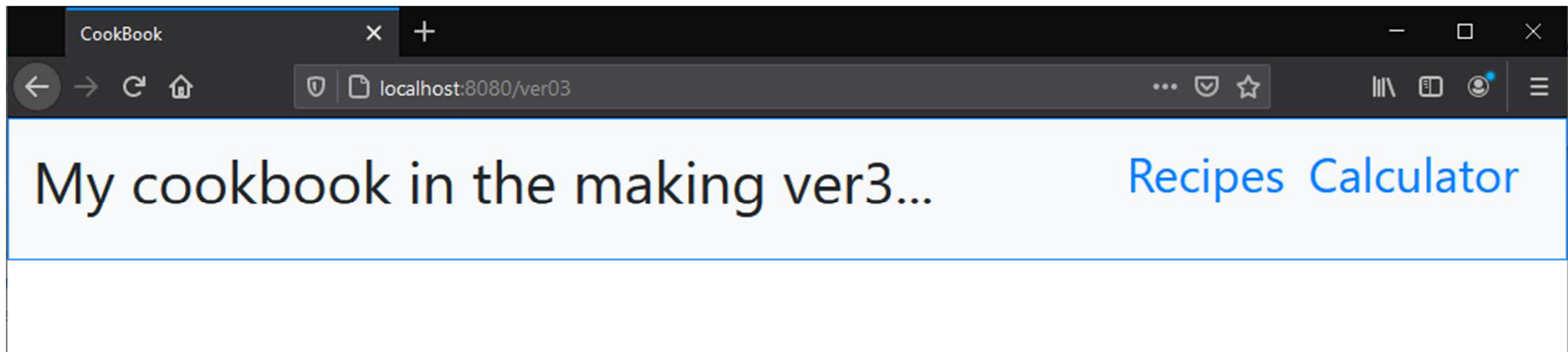
```
let template = `  
<h1>My cookbook in the making ver2 (from the module)...</h1>  
Pick one:  
  <ul>  
    <li><a href="http://fer.unizg.hr">Home</a></li>  
    <li><a data-link href="http://fer.unizg.hr">Abroad</a></li>  
  </ul>  
`;  
  
document.addEventListener("DOMContentLoaded", () => {  
  document.body.addEventListener("click", e => {  
    if (e.target.matches("[data-link]")) {  
      e.preventDefault();  
      document.getElementById("uspa-app")  
        .innerHTML = "Nemoj sine nikud ići...";  
    }  
  });  
  document.getElementById("uspa-app").innerHTML = template;  
});
```

## Ver2: presrećemo klik

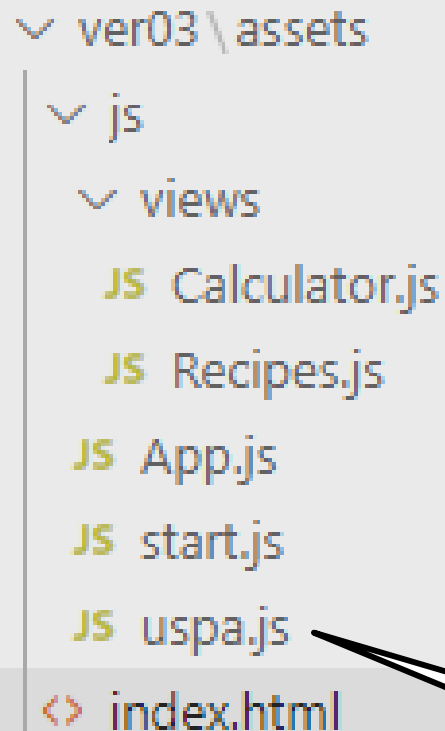
- Pomalo pretjerano, presrećemo svaki klik, pa onda gledamo zanima li nas
  - Mogli smo npr. pronaći samo a elemente pa premostiti njihov click (isprobajte)



- Znamo:
  - Dinamički promijeniti sadržaj stranice
  - Presresti (i promijeniti) klik, npr. kod linkova
- Napravimo:
  - **Višestraničnu** web-aplikaciju – sami upravljamo promjenom stranica!
    - **routing**
  - Pritom preustrojimo kod tako da je svaka stranica poseban **view** odnosno posebna datoteka/modul
    - App view (defaultni view, aplikacija)
    - Recipes view (pregled recepata)
    - Calculator view (preračunavanje C <-> F (u ver4))



# Novi ustroj projekta



ver03 \ assets  
 js  
 views  
 Calculator.js  
 Recipes.js  
 App.js  
 start.js  
 uspa.js  
 index.html

The image shows a file explorer with a tree view. The 'start.js' file is highlighted with a dashed orange box. A line points from this box to a callout box below.

start.js

```
import Uspa from "./uspa.js"
import App from "./App.js"

const app = new Uspa();

app.mount("my-app", new App());
```

FWK je sad posebna datoteka tj. modul



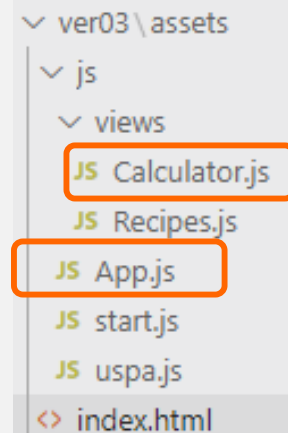
# Svi viewovi moraju imati async getHtml()

## App.js

```
import Recipes from "../views/Recipes.js";
import Calculator from "../views/Calculator.js";
export default class {
  getViews() {
    return {
      Recipes,
      Calculator
    };
  }
  async getHtml() {
    return `
    <div class="d-flex justify-content-between bg-light border border-bottom border-primary p-3">
      <h1>My cookbook in the making ver3...</h1>
      <h2 class="d-flex">
        <ul class="d-flex ml-5 " style="list-style: none;">
          <li class="mr-3"><a data-link href="Recipes">Recipes</a></li>
          <li class="mr-3"><a data-link href="Calculator" >Calculator</a></li>
        </ul>
      </h2>
    </div>
    <router-view></router-view>
    `;
  }
}
```

## Calculator.js

```
export default class {
  async getHtml() {
    return `
    <h1>Calc view</h1>
    `;
  }
}
```



# Uspa.js – nova verzija radnog okvira

uspa.js

```
export default class Uspa {  
  async setView(viewName) {  
    let viewClass = this.currView.getViews()[viewName];  
    document.querySelector("router-view").innerHTML = await (new viewClass()).getHtml();  
    history.pushState(null, null, `${this.stubUrl}/${viewName}`);  
  }  
  mount(selector, initialView) {  
    this.currView = initialView;  
    this.stubUrl = window.location.pathname; // this is only because of server-side ver00 magic  
    initialView.getHtml()  
      .then(html => {  
        document.getElementById(selector).innerHTML = html;  
        return html;  
      }).then(html => {  
        if (html.indexOf("data-link" > 0)) {  
          document.body.addEventListener("click", e => {  
            if (e.target.matches("[data-link]")) {  
              e.preventDefault();  
              let viewName = e.target.getAttribute("href");  
              this.setView(viewName);  
            }  
          });  
        }  
      }).catch(err => {  
        console.log("USPA: error occured:", err);  
      });  
  }  
}
```

History API

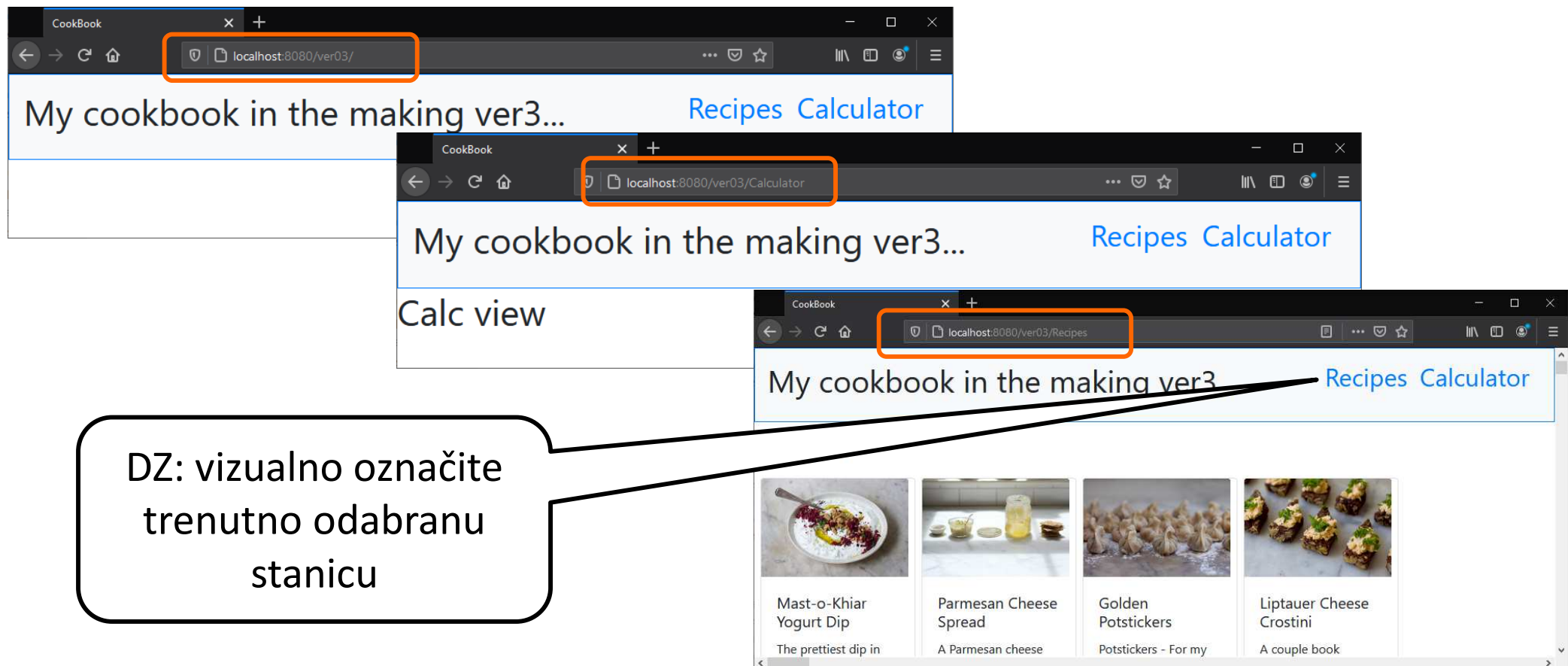
...  
app.mount("my-app", new App());

Npr. „Calculator”

```
ver03\assets  
├── js  
│   ├── views  
│   │   ├── Calculator.js  
│   │   ├── Recipes.js  
│   │   ├── App.js  
│   │   ├── start.js  
│   │   └── uspa.js  
└── index.html
```

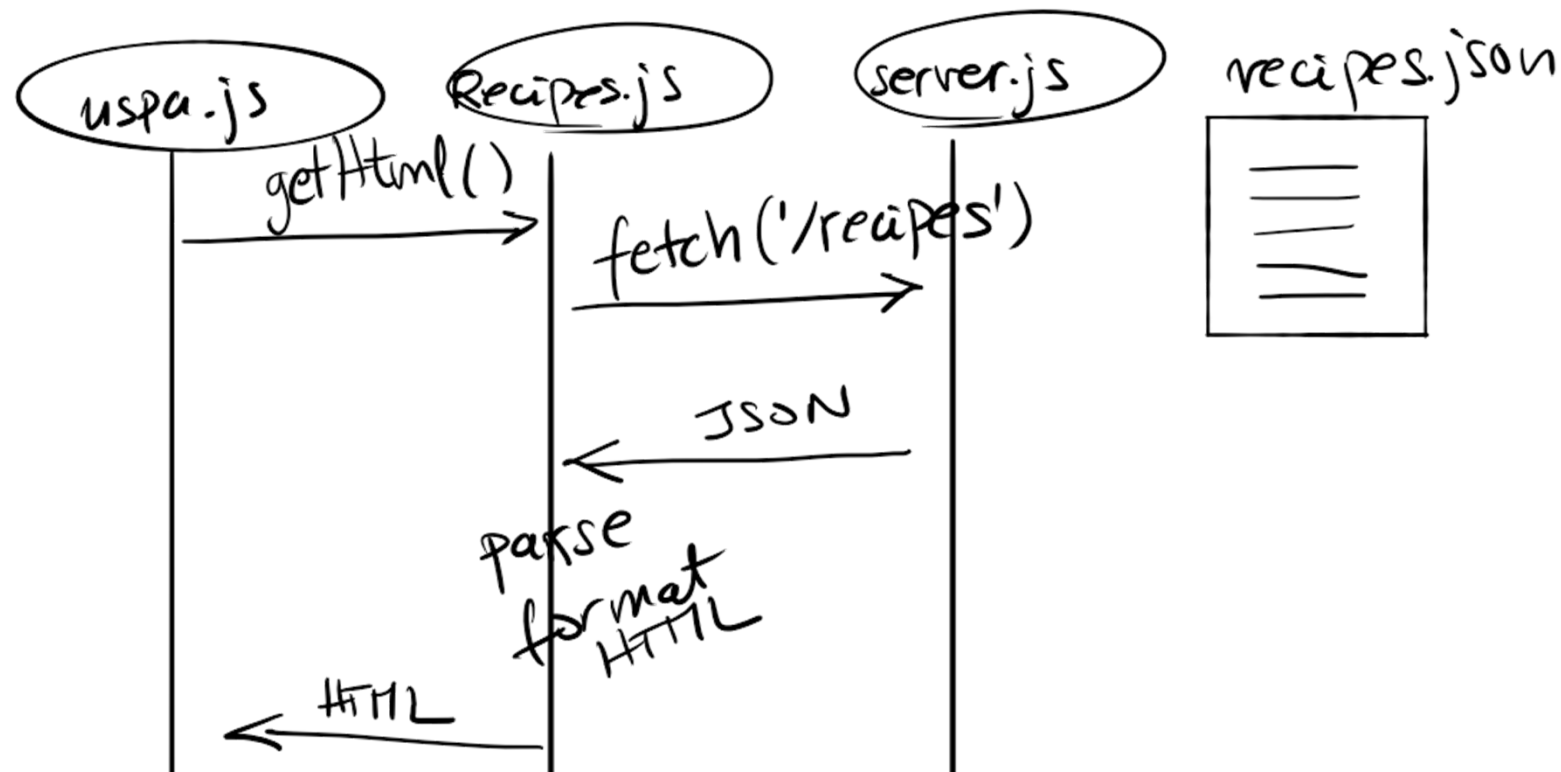
# Višestranična web-aplikacija

- Sve se odvija na klijentu, aplikacija je responzivnija
- Mijenja se URL (s konačnim ciljem: *bookmarking*)
  - Routing nije dalje razvijan, koga zanima kompletniji primjer može pogledati [https://www.youtube.com/watch?v=6BozpmSjk-Y&ab\\_channel=dcode](https://www.youtube.com/watch?v=6BozpmSjk-Y&ab_channel=dcode) ili <https://blog.jeremylikness.com/blog/build-a-spa-site-with-vanillajs/>



# Recipes view

- Dohvatiti ćemo recepte s poslužitelja, formatirati i vratiti HTML
- Sad je jasno zašto `getHtml()` mora biti **async**
- **Uobičajen obrazac kod SPA – dohvat s poslužitelja putem Ajax poziva**



# Pogledajmo prvo server.js (iako je van fokusa)

server.js

```
const fs = require('fs');
let jsonRecipes;
fs.readFile('recipes.json',
  function (err, data) {
    var jsonData = data;
    jsonRecipes = JSON.parse(jsonData);
    console.log(jsonRecipes.length + " recipes parsed...");
  });

...

app.get('/recipes', function (req, res) {
  res.json(jsonRecipes);
});
```

# recipes.json

recipes.js

Preuzeto s:

<https://github.com/fictivekin/openrecipes>

```
[{
  "name": "Easter Leftover Sandwich",
  "ingredients": "12 whole Hard Boiled Eggs\n1/2 cup Mayonnaise\n3 Tablespoons Grainy Dijon Mustard\nSalt And Pepper, to taste\nSeveral Dashes Worcestershire Sauce\nLeftover Baked Ham, Sliced\nKaiser Rolls Or Other Bread\nExtra Mayonnaise And Dijon, For Spreading\nSwiss Cheese Or Other Cheese Slices\nThinly Sliced Red Onion\nAvocado Slices\nSliced Tomatoes\nLettuce, Spinach, Or Arugula",
  "url": "http://thepioneerwoman.com/cooking/2013/04/easter-leftover-sandwich/",
  "image": "http://static.thepioneerwoman.com/cooking/files/2013/03/leftoversandwich.jpg",
  "cookTime": "PT",
  "recipeYield": "8",
  "datePublished": "2013-04-01",
  "prepTime": "PT15M",
  "description": "Got leftover Easter eggs? Got leftover Easter ham? Got a hearty appetite? Good! You've come to the right place! I..."
},
{
  "name": "Pasta with Pesto Cream Sauce",
  "ingredients": "3/4 cups Fresh Basil Leaves\n1/2 cup Grated Parmesan Cheese\n3 Tablespoons Pine Nuts\n2 cloves Garlic, Peeled\nSalt And Pepper, to taste\n1/3 cup Extra Virgin Olive Oil\n1/2 cup Heavy Cream\n2 Tablespoons Butter\n1/4 cup Grated Parmesan (additional)\n12 ounces, weight Pasta (cavatappi, Fusili, Etc.)\n2 whole Tomatoes, Diced",
  "url": "http://thepioneerwoman.com/cooking/2011/06/pasta-with-pesto-cream-sauce/",
  "image": "http://static.thepioneerwoman.com/cooking/files/2011/06/pesto.jpg",
  "cookTime": "PT10M",
  "recipeYield": "8",
  "datePublished": "2011-06-06",
  "prepTime": "PT6M",
  "description": "I finally have basil in my garden. Basil I can use. This is a huge development. I had no basil during the winter. None. G..."
}
```

```
export default class {
  async getHtml() {
    try {
      let response = await fetch('/recipes');
      if (response.ok) {
        let allRecipes = await response.json();
        let recipes = allRecipes.filter(x => Math.random() > 0.75).reverse();
        return `<h1>Recipes (${recipes.length})</h1>`
          + `<div class="container-fluid p-2 d-flex flex-wrap">`
          + recipes.map(rcp => `
            <div class="card mt-2 mr-2" style="width: 200px;">
            
            <div class="card-body">
              <h5 class="card-title">${rcp.name}</h5>
              <p class="card-text">${rcp.description}</p>
              <span class="badge badge-primary">
                Cook/prep time: ${rcp.cookTime}/${rcp.prepTime}</span>
                (...izbačen dio kod koji formatira...)
            </div>
          `).join('') + `</div>`;
      } else { throw new Error("HTTP-Error: " + response.status); }
    } catch (error) {
      return `<h1>Home view (error occured)</h1>`;
    }
  }
}
```

Samo ~četvrtinu slučajno  
odabranih (ima ih 1042)



## My cookbook in the making ver3...

## Recipes Calculator

## Recipes (274)



## Gougères

Gougères - I have these little cheese puffs in my freezer, ready to bake, nearly always. This version, a favorite, is made with whole wheat flour, sharp white cheddar cheese, fennel, and ale.

Cook/prep time: PT30M/PT10M

Yield:

Published: 2011-12-17

[Source](#)

Ingredients



## Grilled Salt &amp; Vinegar Potatoes

Salt & vinegar chip enthusiasts, these are for you. You take slabs of sliced potatoes, boil them in vinegar, then grill them to a crisp. Not for the faint of heart, or anyone with particularly sensitive taste buds ;)...

Cook/prep time: PT10M/PT35M

Yield:

Published: 2010-07-11

[Source](#)

Ingredients



## Baked Sweet Potato Falafel Recipe

Baked Sweet Potato Falafel recipe from the Leon cookbook -made from mashed sweet potatoes, chickpea flour, spices, a nice amount of garlic and plenty of chopped cilantro.

Cook/prep time: / Yield:

Published: 2009-05-17

[Source](#)

Ingredients



## Classic Cheese Fondue Recipe

A classic cheese fondue recipe. I've included dozens of my favorite things to dunk - don't limit yourself to just bread!

Cook/prep time: / Yield:

Published: 2008-12-26

[Source](#)

Ingredients



## Caramelized Onion Dip Recipe

A grown-up remake of the onion dip I loved as a kid. This one features lots of deeply caramelized onions, sour cream and Greek yogurt.

Cook/prep time: /

Yield: Makes about 2 cups.

Published: 2008-12-06

[Source](#)

Ingredients



## Hummus en Fuego Recipe

A beautiful, spicy hummus recipe made from pureed garbanzo beans, toasted walnuts, and spicy crushed red pepper oil finished with a few chopped olives and a bit of cilantro.

Cook/prep time: /

Yield: Makes roughly 2 1/2 cups...

Published: 2008-10-08

[Source](#)

Ingredients

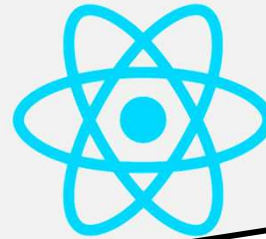


## Ver 3 osvrt

- ✓ Razložili smo aplikaciju na viewove
  - Pravi radni okviri će omogućiti i manje gradivne blokove - komponente
  - View se brine samo za sebe i stvaranje svog HTML-a
- ✓ (Proto)usmjeravanje, mijenjamo stranice na klijentu, kozmetički mijenjamo URL
- ✗ Loše:
  - ✗ Kako formiramo HTML – string concatenation, spaghetti code
- Treba nam:
  - Templating „engine”, elegantan sustav kako ćemo napraviti HTML

# Primjer: stvaranje HTML-a: React i Vue

```
class App extends React.Component {  
  render() {  
    const planets = ['Merkur', 'Venera', 'Zemlja'];  
    return (  
      <ol>  
        {planets.length > 0 && planets.map(n => (  
          <li key={"section-" + n}>Planet {n}</li>  
        ))}  
      </ol>  
    );  
  }  
}
```



JSX, or JavaScript XML, is an extension to the JavaScript language syntax

```
<template>  
  <div id="app">  
    <ol>  
      <li v-for="planet in planets">Planet {{ planet }}</li>  
    </ol>  
  </div>  
</template>  
<script>  
export default {  
  data() {  
    return {  
      planets: ['Merkur', 'Venera', 'Zemlja']  
    }  
  }  
}; </script>
```



```
<ol>  
  <li>Planet Merkur</li>  
  <li>Planet Venera</li>  
  <li>Planet Zemlja</li>  
</ol>
```

Ni jedno ni drugo nije validan JS, treba prevesti...



# V4 u sljedećem predavanju...